Q1. Which two operator overloading methods can you use in your classes to support iteration?

+= and ++

Q2. In what contexts do the two operator overloading methods manage printing?

Q3. In a class, how do you intercept slice operations?

 the\_\_setitem\_index assignment method similarly intercepts both index and slice assignments—it receives a slice object for the latter, which may be passed along in another index assignment in the same way

Prior to Python 3.0, classes could also define\_getslice\_and\_setslice\_methods to intercept slice fetches and assignments specifically; they were passed the bounds of the slice expression and were preferred over\_getitem\_and\_setitem\_for slices.

Q4. In a class, how do you capture in-place addition?

 iadd() :- This function is used to assign and add the current value. This operation does “a+=b” operation. Assigning is not performed in case of immutable containers, such as strings, numbers and tuples.

Q5. When is it appropriate to use operator overloading?

It allows us to provide an intuitive interface to our class users, plus makes it possible for templates to work equally well with classes and built-in types. Operator overloading allows python operators to have user-defined meanings on user-defined types or classes.